01/22/25

Hash Tables

Python dictionary (like Java map)

Hash function:

- When given same input, will produce same output
- Output of the hash function will give the location/index of the bucket
    - Last step will be mod table size to guarantee that only possible indexes will be given
- Requires constant work for any input
    - (With empty slots) constant work to add values to array, and constant work to look up values
    - With separate chaining (and with a good hash function), searching the chains is “essentially” constant time
        - If you have a lot more slots than inserted values, then the probability that the chains are long is low
- Good dispersion: hash function should spread the values out instead of putting them in the same slots
- Used to determine the index for insertion and look up

Table size

- Start with a big table
- Use load factor to determine when we have put too many values in the table and need to make a bigger one
    - Load factor should be less than 0.9
    - Rehashing: when you make a bigger table, you have to reindex every value
- Why not make biggest table you can?
    - You get data as it comes and need to change the table in response

    - Space efficiency

Load factor (λ): how many values have been inserted over how many slots there are (table size)

Conflicts

- Separate chaining: if two values are hashed to the same location, then the location becomes a bucket that has a list of all the values that are hashed to that slot
    - The list can also be a set or other data types
    - The list is not sorted